## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
## BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Patent Application No. 09/954,508

Confirmation No. 8453

Applicant: Todorov et al.

Filed: September 14, 2001

TC/AU: 2144

Examiner: Nguyen, Thanh T.

Docket No.: 211626 (Client Reference 02,214 US)

Customer No.: 23460

## APPELLANTS' REPLY UNDER 37 C.F.R. SECTION 41.41

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This paper is filed in response to the Examiner's Answer mailed on **October 4, 2007**. Appellants hereby respectfully request allowance of the pending claims for the reasons set forth in Appellants' Brief filed on **June 29, 2007**, and for the further reasons stated herein.

*Status of Claims*

Claims 1-50 stand finally rejected, and these rejections are presently being appealed.

A complete listing of these claims appears in the Claims Appendix.

*Grounds of Rejection to be Reviewed On Appeal*

The grounds of rejection to be reviewed on appeal are the grounds stated in the Final Office Action mailed on August 9, 2006. In particular, Appellants appeal the rejection of Claims 1-50 as obvious under 35 U.S.C. Section 103(a) over Dorrance et al., U.S. Patent No. 6,430,598 (Dorrance) in view of Lim et al., U.S. Patent No. 6,718,550 (Lim).

*Argument in Reply to the Answer*

Appellants have carefully reviewed the Examiner's Answer. The Grounds of Rejection Section (9) of the Answer, beginning at page 3 and ending at page 17, restates the grounds recited in the Final Office Action from which the present appeal was taken. Appellants have addressed the Answer's grounds for rejection in the previously filed appeal brief, and thus Appellants will not repeat these arguments. However, the discussion below reiterates the points previously made by Appellants in order to address the points raised in the Answer's Response to Argument Section (10).

Appellants seek reversal of the rejection of the presently pending claims (provided in the Claims Appendix attached hereto). The claimed invention is directed to a server that provides clients access to process data according to multiple differing data exchange protocols. The multiple data exchange protocol support is facilitated by a *set of client data exchange protocol modules* (note the plural term "modules"). The *set* of client data exchange protocol modules convert client protocol-specific requests into a generic set of calls supported by a *data exchange protocol abstraction layer* within a data access server engine.

Neither Dorrance nor Lim disclose the aforementioned *set of client data exchange protocol modules* and *data exchange protocol abstraction layer*. Dorrance discloses an email server that supports multiple email protocol requests through a *single* converter 65 entity. Lim does indeed disclose a set of clients. However, the presence of a set of clients does not overcome a complete absence of any teaching in either Dorrance or Lim with regard to a process data access server component arrangement wherein a *set* of protocol-specific client modules receive protocol-specific client requests and issue corresponding operation calls supported by a *data exchange protocol abstraction layer* of a data access server engine.

The Response to Argument Section (10) does not appear to raise any significant new issues or arguments, and instead summarizes the erroneous grounds already presented in Section (9) that were addressed in Appellants' Appeal Brief. Nevertheless, Appellants address the issues presented in Section (10) of the Examiner's Answer herein below.

- *Dorrance Does Not Disclose the claimed "Set of Client Data Exchange Modules" and "Protocol Abstraction Layer"*

The Answer's Response, at page 17, to Appellants' Argument initially asserts that Applicant's argument is vague. Appellants' submit that their brief unequivocally demonstrates the complete absence of a teaching in Dorrance of a *set* of client data exchange protocol *modules* and a protocol abstraction layer in the data access server engine that receives requests from the modules. Appellants' claims define a process data access server comprising a *modular, multi-component, architecture that supports multiple data exchange protocols.* Dorrance, rather than disclosing Appellants' claimed multiple module approach for supporting multiple data exchange protocols, discloses a converter 65 (see, FIG. 3) consisting of a *single* functional component. There is no specific teaching within Dorrance of Appellants' claimed *set of client data exchange protocol modules* and *protocol abstraction layer* (within a data access server engine) including a set of operations callable by the protocol modules. **Appellants request identification of the specific elements in Dorrance's converter 65 and server 62 that correspond to these distinctly identified components of Appellants' claimed process data access server.**

- *Dorrance Does Not Disclose a Process Data Access Server*

Responding to the Answer's response at pages 17 and 18, Appellants again note that Dorrance discloses an *email server* that has absolutely no relation to the claimed "process data access server" which operates in a process control/manufacturing/plant environment to provide access to status/control information associated with a manufacturing/industrial process. **Appellants strongly suggest a careful review of column 6, lines 6-14 that merely identify the initiation of the *process* of handling a received email server request from a client.** Dorrance does not even remotely disclose the claimed "process data access server."

- *Dorrance Does Not Disclose "Plugins"*

The Response, at page 18, identifies column 4, lines 1-20 as disclosing Plugins. However, there is no teaching whatsoever of the Plugin programming construct recited in claims 2-5, 30, and 45-48.

- *Dorrance Does Not Disclose a Set of Protocol-Specific Modules*

Appellants are unsure of the portion of their Brief to which the Examiner's Response is directed (see, second bullet point on page 18 of the Examiner's Answer). However, Appellants addressed the absence of any teaching of the claimed set of modules in Dorrance herein above. Rather than describe Appellants' claimed set of modules for supporting a set of data exchange protocols, Dorrance discloses a *single* converter (65) component.

- *"Appellant Argues That Dorrance Does Not Disclose Converters"*

Appellants are unable to determine the source for this quote from the third bullet point on page 18 of the Answer since there is no citation to either a claim or page number of Appellants' Brief. Applicants have previously stated that Dorrance discloses a *single* converter 65 rather than a set of modules associated with particular data exchange protocols supported by a process data access server.

The Response to Argument Section (10) does not address several of Appellants' arguments in their Appeal Brief, and it is assumed that the Examiner intends to rely upon the grounds stated in Section (9). The grounds recited in Section (9) are identical to the grounds presented in the Final Office Action. Appellants have addressed those grounds in their previously filed Appeal Brief, and thus Appellants will not repeat these arguments here.

*Conclusion*

In summary, the present invention is distinguishable from the combined teachings of the cited references for a variety of reasons. The invention recited in the presently pending claims is directed to a process data access server that supports a variety of client data exchange protocols (e.g., DDE, OPC, SuiteLink, etc.) via a set of data exchange protocol modules. The multiple protocol modules interface with a data access server engine via an abstraction layer comprising a set of callable operations. While the prior art does indeed disclose supporting multiple protocols, the recited *way* in which multiple client data exchange protocol support is provided in Appellants' claimed invention is neither disclosed nor suggested in the prior art. For at least the reasons set forth in Appellants' Brief and herein above, each of the presently pending claims is patentable over the prior art.

Appellants therefore request reversal of the presently pending rejection of claims 1-50.

Respectfully submitted,

Mark Joy, Reg. No. 35,562
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza
180 North Stetson Ave., Suite 4900
Chicago, Illinois 60601-6731
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: December 4, 2007

*Claims Appendix*

1. (Original) A process data access server enabling client applications incorporating potentially multiple differing data exchange protocols to access process data stored at potentially many different locations in a process control system, the process data access server comprising:

a device protocol interface facilitating accessing process data storage locations within the process control system;

a set of client data exchange protocol modules enabling client applications to request access to process data storage locations via the process data access server according to particular client data exchange protocols supported by the set of client data exchange protocol modules; and

a data access server engine for executing process data access requests, received by the process data access server via the set of client data exchange protocol modules, by accessing, via the device protocol interface, data storage locations corresponding to the process data access requests, and wherein the data access server engine includes a client application data exchange protocol abstraction layer comprising a set of operations callable by ones of the set of client data exchange protocol modules in response to receipt by the set of client data exchange protocol modules of process data access requests.

2. (Original) The process data access server of claim 1 wherein the set of client data exchange protocol modules comprise plugins.

3. (Original) The process data access server of claim 2 wherein at least one of the set of client data exchange protocol plugins comprises a dynamic plugin.

4. (Original) The process data access server of claim 2 wherein at least one of the set of client data exchange protocol plugins comprises a static plugin.

5. (Original) The process data access server of claim 2 wherein the set of protocol conversion modules comprise both static and dynamic plugins.

6. (Original) The process data access server of claim 1 wherein ones of the set of client data exchange protocol modules handle data access requests from client applications in accordance with particular client data exchange protocols.

7. (Original) The process data access server of claim 1 further including:

a loading mechanism for determining a presence of at least one of the set of client data exchange protocol modules upon a machine for executing the process data access server, and loading the at least one client data exchange protocol module during a startup process that integrates the at least one client data exchange module with the data access server engine.

8. (Original) The process data access server of claim 1 wherein the set of operations of the data access server engine includes at least one operation callable by at least two distinct ones of the set of client data exchange protocol modules that incorporate distinct data exchange protocols.

9. (Original) The process data access server of claim 1 wherein an operational data access server including the device protocol interface, the set of client data exchange protocol modules, and the data access server is created by a start-up process that builds the operational data access server from previously installed program files, and wherein the program files of the client data exchange protocol modules and the data access server are independently designateable with regard of one another.

10. (Original) The process data access server of claim 1 wherein the set of interface operations executable by the data access server engine includes an asynchronous data read operation for providing data from an identified data source in response to a client application data request.

11. (Original) The process data access server of claim 1 wherein the set of interface operations executable by the data access server engine includes a synchronous read operation that, in accordance with a timer duration expiration event, updates identified process data values via the device protocol interface.

12. (Original) The process data access server of claim 11 wherein the synchronous read operation discards an updated process data value for a data item that is determined to be unchanged from a current stored value for the data item, thereby avoiding transmissions of unchanged data values between the process data access server and requesting client applications.

13. (Original) The process data access server of claim 1 wherein the set of interface operations executable by the data access server engine includes a group creation operation that creates a first logical group containing a first set of data items.

14. (Original) The process data access server of claim 13 wherein a second logical group containing a second set of data items is includable as an item within the first logical group containing the first set of data items.

15. (Original) The process data access server of claim 13 wherein the set of interface operations executable by the data access server engine includes a group remove operation that removes a specified group from the process data access server.

16. (Original) The process data access server of claim 13 wherein the set of interface operations executable by the data access server engine includes operations for modifying the contents of the first logical group.

17. (Original) The process data access server of claim 1 wherein the set of interface operations executable by the data access server engine includes a write operation to a specified data item accessible by the process data access server.

18. (Original) The process data access server of claim 1 wherein the set of interface operations includes a data reference structure search operation that returns a data item reference corresponding to a data item value accessible by the client applications via the process data access server.

19. (Original) The process data access server of claim 18 wherein the data item reference is a handle.

20. (Original) The process data access server of claim 1 wherein the set of interface operations includes an error code generator that supplies error code text to a requesting client data exchange protocol module.

21. (Original) The process data access server of claim 1 wherein the set of interface operations includes a status reporter operation that provides access to a data structure that stores status values for the process data access server.

22. (Original) A method for providing, by a data access server, access to process data in a distributed process control environment in accordance with a client application data exchange protocol supported by one of a set of client application data exchange protocol modules installed on the data access server, and wherein the set of client application data exchange protocol modules invoke a set of data access operations executable by a data access server engine of the data access server according to a module-engine interface definition, the method comprising the steps of:

receiving, by a first client application data exchange protocol module of the data access server, a first client application data access request according to a first data exchange protocol;

first generating, by the first client application data exchange protocol module, a first data access operation call for the data access server engine conforming to the module-engine interface definition, wherein the first data access operation call corresponds to the first client application data access request; and

executing, by the data access server engine, the first data access operation call.

23. (Original) The method of claim 22 further comprising the steps of:

second generating, by the data access server engine, a response to the first data access operation call; and

third generating, by the first data exchange protocol module, a response to the first client application data access request, wherein the response to the first client application data access request corresponds to the response to the first data access operation call generated by the data access server engine during the second generating step.

24. (Original) The method of claim 22 further comprising the steps of:

second receiving, by a second client application data exchange protocol module of the data access server, a second client application data access request according to a second data exchange protocol; and

second generating, by the second client application data exchange protocol module, a second data access operation call for the data access server engine conforming to the module-engine interface definition, wherein the second data access operation call corresponds to the second client application data access request.

25. (Original) The method of claim 24 wherein the first data access operation call is identical to the second data access operation call.

26. (Original) The method of claim 22 further comprising the step of:

receiving, by the first client application data exchange protocol module, a request to create a logical group that contains a set of data items representing data accessed in the process control environment, and a further request to add a data item to the logical group.

27. (Original) The method of claim 26 wherein the first client application data access request comprises a subscription query requesting the data access server to issue a notification in response to detecting a change to a data value associated with the data item within the logical group.

28. (Original) The method of claim 27 wherein the executing step comprises forwarding a request for device data to a device protocol interface, and wherein the device protocol interface transmits a corresponding data request to a field device according to a field device-specific request protocol.

29. (Original) The method of claim 28 further comprising the steps of:

receiving, by the device protocol interface, a response from the field device comprising data corresponding to the data item;

forwarding, by the device protocol interface to the data access server engine, a response message including a data value for the data item.

30. (Original) The method of claim 22 wherein the set of client data exchange protocol modules comprise plugins.

31. (Original) The method of claim 22 wherein ones of the set of client data exchange protocol modules handle data access requests from client applications in accordance with particular client data exchange protocols.

32. (Original) The method of claim 22 wherein the executing step comprises performing an asynchronous data read operation for providing data from an identified data source in response to a client application data request.

33. (Original) The method of claim 22 wherein the executing step comprises performing a synchronous read operation that, in accordance with a timer duration expiration event, updates identified process data values via the device protocol interface.

34. (Original) The method of claim 33 wherein the synchronous read operation discards an updated process data value for a data item that is determined to be unchanged from a current stored value for the data item, thereby avoiding transmissions of unchanged data values between the process data access server and requesting client applications.

35. (Original) The method of claim 22 wherein the first data access operation call comprises a group creation operation that creates a first group containing a first set of data items.

36. (Original) The method of claim 35 further comprising executing, by the data access server engine, a second data access operation call that adds a second logical group containing a second set of data items as a group item within the first group.

37. (Original) The method of claim 35 further comprising executing, by the data access server engine, a second data access operation call that removes a specified group from the data access server.

38. (Original) The method of claim 35 further comprising executing, by the data access server engine, a second data access operation call to modify contents of the first logical group.

39. (Original) The method of claim 22 wherein the first data access operation call comprises a write operation to a specified data item accessible by the data access server.

40. (Original) The method of claim 22 wherein the first data access operation call comprises a data reference structure search operation that returns a data item reference corresponding to a data item value accessible by the client applications via the process data access server.

41. (Original) The method of claim 40 wherein the data item reference is a handle.

42. (Original) The method of claim 22 wherein the first data access operation call comprises an error code generator operation that supplies error code text to the requesting client data exchange protocol module.

43. (Original) The method of claim 22 wherein the first data access operation call comprises a status reporter operation that provides access to a data structure that stores status values for the data access server.

44. (Original) A method for activating a data access server through a start-up process that builds the data access server from previously installed program files including at least an executable file incorporating a data access server engine and a separate and distinct file containing one or more of a set of client application data exchange protocol modules installed on the data access server, and wherein the set of client application data exchange protocol modules invoke a set of data access operations executable by the data access server engine of the data access server according to a module-engine interface definition, the method comprising the steps of:

starting up an executable corresponding to the data access server and including the data access server engine;

loading the set of client application data exchange protocol modules thereby creating program links between at least one of the protocol modules and the data access server executable; and

instantiating a data access server object corresponding to a connection between the data access server and a requesting client application.

45. (Original) The method of claim 44 wherein the set of client data exchange protocol modules comprise plugins.

46. (Original) The method of claim 45 wherein at least one of the set of client data exchange protocol plugins comprises a dynamic plugin.

47. (Original) The method of claim 45 wherein at least one of the set of client data exchange protocol plugins comprises a static plugin.

48. (Original) The method of claim 45 wherein the set of protocol conversion modules comprise both static and dynamic plugins.

49. (Original) The method of claim 44 wherein ones of the set of client data exchange protocol modules handle data access requests from client applications in accordance with particular client data exchange protocols.

50. (Original) The method of claim 44 wherein the set of operations of the data access server engine includes at least one data access operation callable by at least two distinct ones of the set of client data exchange protocol modules that incorporate distinct data exchange protocols.